Modified Cryptanalysis of RC5

Mowafak Hasan and Hasan Al-Shalabi
College of Computer Engineering and Information Technology, Al Hussein Bin Talal University, Jordan

Abstract: *The RC5 encryption algorithm was designed by Roland Rivest in 1994. Since its publication, RC5 has attracted the attention of many researchers in the cryptographic community in efforts to accurately assess the security offered. The best previously known attack requires 2^{54} chosen plaintexts in order to derive the full set of 25 subkeys for the 12 round RC5 with 32 bit words. In this paper, we modified these results due to a differential approach. The attack requires 2^{34} chosen plaintexts. We show that the 64 bit word version of RC5 is also weaker than it was expected.*

Keywords: *RC5, cryptographic, differential cryptanalysis.*

Received March 31, 2005; accepted May 302005

1. Introduction

RC5 is a symmetric encryption algorithm developed by Roland Rivest [7], it was designed to have the following objectives [8]:

- *Symmetric block cipher*: The same secret cryptographic key is used for encryption and for decryption. The plaintext and ciphertext are fixed–length bit sequences.
- *Suitable for hardware or software*: It should use only primitive computational operations commonly found on typical microprocessors.
- *Fast*: RC5 is a simple algorithm and is word oriented. The basic operations work on full words of data at a time.
- *Adaptable to processors of different word– length*: The number of bit in a word is a parameter of RC5; different word lengths yield different algorithms.
- *Variable number of rounds*: The number of rounds is a second parameter of RC5. This parameter allows a tradeoff between higher speed and higher security.
- *Variable–length cryptographic key*: It is the third parameter of RC5. Again, this allows a tradeoff between speed and security.
- *Simple*: RC5's simple structure is easy to implement and eases the task of determining the strength of the algorithm.
- *Low memory requirement*: This property makes the algorithm suitable for smart cards and other devices with restricted memory.
- *High security*: It should provide high security when suitable parameter values are chosen.
- *Data–dependent rotations*: RC5 incorporates rotations (circular bit shifts) whose amount is data dependent. This indicates to strengthen the algorithm against cryptanalysis.

The modification of the RC5 encryption algorithm presented here hopefully meets all of the above objects.

2. Description and Features of RC5

RC5 is a parameterized algorithm, and a particular RC5 is designated as RC5–w/r/b. Table 1 summarized these three parameters. The "nominal" choice of the parameters proposed in [6] is RC5–32/12/16. Another version with 64 bit words and 16 rounds was suggested for future 64 bit architectures (RC5–64/16/16).

Table 1. Parameters of RC5.

Parameter	Definition	Allowable Values
w	Word size in bit, RC5 encrypts 2 – word blocks	16, 32, 64, 128
r	Number of rounds	0, 1, … ,255
b	Number of 8–bit bytes (Octal) in the secrete key K	0, 1, … ,255

The application of the two powerful attacks of differential linear cryptanalysis to RC5 is considered by Kaliski and Yin [2] who evaluate that the 12- round nominal cipher appears to be secure against both attacks. In [3], Knudsen and Meier extend the analysis of the differential attacks of RC5 and perform that the complexity can be reduced by a factor of up to 512 for a typical key of the nominal RC5. Recently, in [4], new differential cryptanalysis result imply that 16 round are required for the cipher with w = 32 to be secure Kocher [5] introduces the general notion of a timing attack, which reveal the key by making use of information or the time it takes to encrypt.

3. Notations and RC5 Primitive Operations

The RC5 algorithms use the following three primitive operations (and their inverse):

1. Bit-wise exclusive -OR of words, denoted by ⊕.
2. The inverse operation, NOR, is denoted by "-".
3. Rotation: The rotation of x to the left by y bits is denoted by x <<< y.

Note that we use $\lg_2$ (x) to denote the base-two logarithm of x.

To decrypt, the operations of the algorithm must be appropriately reversed to generate the data for each half–round by essentially going back wards through the encryption algorithm.

All operations are directly and efficiently supported by most processors.

3.1. Key Expansion

The key–expansion algorithm expands the user's key K to fill the expanded key table S, so that S resembles an array of t = 2 (r + 1) random binary words determined by K. Figure 1 illustrates the technique used to generate subkeys.

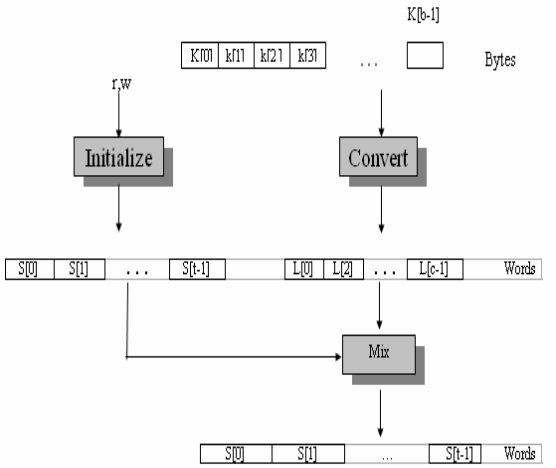


Figure 1. Key encryption.

The subkeys are stored in a t-word array S [0 .. t - 1]. Using the parameters r and w as inputs, S is initialized to a particular fixed pseudorandom bit pattern. Then the b–bytes key, K [0], K [1], …, K [b - 1], is converted into a c–word array L [0], L [1], …, L [c - 1], where c = [b / u] words, where u = w / 8 is the number of by word.

The initialize operation makes of two magic constants P_W and Q_W defined for arbitrary w as follows:

$$P_W = Odd((e-2)\,2^W)$$
$$Q_W = Odd((\phi-1)\,2^W)$$

Where:

e = 20718281828459 … (base of natural logarithm).

ϕ = 1.618033988749… (golden ration = $\frac{1+\sqrt{5}}{2}$) and

where *Odd(x)* is the odd integer nearest to *x* (rounded up if *x* is an even integer. Although this won't happen here). For W = 16, 32, and 64, these constants are given in Table 2.

Table 2. Constant values.

w	16	32	64
P_W	b7el	b7e15163	b7el51628aed2a6b
Q_W	ge37	ge3779b9	ge3779b97f4a7c15

The initialization of the array S, with the two constants P_W and Q_W, can be deduced by the following algorithm:

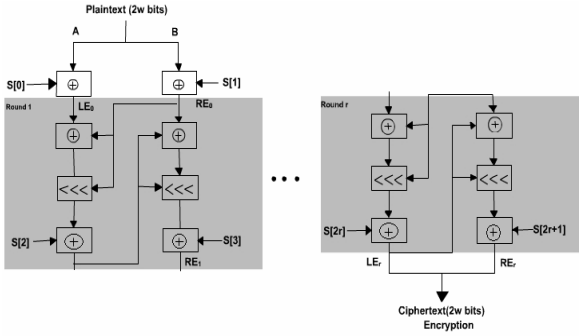
Algorithm

Input: The constants P_W and Q_W, the parameters t.

$S[0] = P_W$

For $i \leftarrow 1$ *to* $t-1$ *do*

$S[i] \leftarrow S[i-1] \oplus Q_W$



The initialization array S is there mixed with the key array L to produce a final array S of subkeys. More precisely the larger array will be processed three time, and the other array may be handled more times.

$i = j = A = B = 0$

Do $3 * max(t,c)$ *times*

$A = S[i] = (S[i] \oplus A \oplus B) <<< 3$

$B = L[j] = (L[j] \oplus A \oplus B) <<< (A \oplus B)$

$i = (i+1) \bmod (t)$

$j = (j+1) \bmod (c)$

Rivest [7] comments that the key–expansion function has a certain amount of "one–wayness": It is not so easy to determine K from S.

3.2. Encryption and Decryption

We use the notation LE_i and RE_i to represent the left and right half respectively of the cipher data at the input to the ith half round. Letting LE_i and RE_i represent the left and right half of the ciphertext respectively, the modified RC5 encryption algorithm is then given by:

$LE_1 = A \oplus S[0]$

$RE_1 = B \oplus S[1]$

For $i = 1$ *to* $2r$ *do*

$LE_{i+1} = RE_i$

$RE_{i+1} = ((LE_i \oplus RE_i) <<< r_i(K)) \oplus S[i+1]$

Where r_i *(K)* is a rotation amount derived from the secret key K. The decryption routine is easily derived from the encryption routine. Figure 2 depicts the encryption operation:

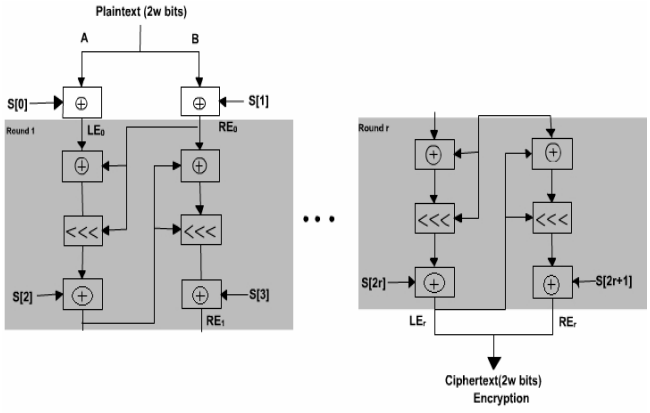


Figure 2. Encryption operation.

4. Experimental Results on the Modification

In this section, we develop a new method to determine a theoretical estimate for "good pairs" probability for RC5 ⊕, roughly like a Fibonacci sequences

$$0,1,1,2,3,5,8,13,21,34,55, \ ...$$

we noticed that in a good pair the weights of the new method behave like

$$0\ ,1\ ,1,2,1,3,2,5,3,8,5,13,8,21,13,34,21,55, \ ...$$

which we called it the modified Fibonacci sequence with an alternative operation respectively. Therefore we can define:

Definition: A sequence of numbers F (n) is called modified Fibonacci sequence with k modifications if the formula:

$$F(i) = F(i-1) + F(i-2);$$
$$i = 2, 3, 4, 5, \ldots$$

is exchange to

$$F(2i) = F(2i-1) - F(2i-2)$$
$$F(2i+1) = F(2i) + F(2i-1);$$
$$\text{For } \ i = 1, 2, 3, 4, 5, \ldots$$

Which is also exactly k times.

We generalized all possible Fibonacci sequences for all reasonable numbers of modification and calculated the probabilities of patterns to such sequences. Table 3 indicates that the upper bounds for the probability of a good pair for RC5 ⊕ (with 16, 32, and 64 bit words), is equivalent to the sum of the probabilities of all possible modified Fibonacci sequence of length n = 2r. The following table gives also the theoretical estimate of good pair probability:

Table 3. Lower and upper bound.

r	12	13	14	15	16
32	$2^{-24.0}$	$2^{-27.0}$	$2^{-29.4}$	$2^{-31.8}$	$2^{-34.2}$
64	$2^{-30.1}$	$2^{-33.2}$	$2^{-36.6}$	$2^{-40.2}$	$2^{-40.1}$

Moreover, the table also indicates that the behavior of the sequence changes with increasing number of rounds, while for a large number of rounds the behavior of the sequence will appear as a concatenation of iterative sequence

$$0\ 1\ 1\ 2\ 1\ 3\ 2\ 5\ 3\ 8$$

This feature can be used to construct an efficient filter of good pair. Following [1] the attack can use successfully several linear independent input differences β_i ; $i = 1, ..., k$, which require the ciphertexts of A as:

$$A \oplus \beta_1, A \oplus \beta_2, \ldots, A \oplus \beta_1 \oplus \beta_2, A \oplus \beta_1 \oplus \beta_3,$$
$$\ldots, A \oplus \beta_1 \oplus \beta_2 \oplus \beta_3, \ \ldots$$

Then a pool of 2^k such ciphertexts contains $k.2^k$ pairs with differences from the set $\{\beta_1, \beta_2, \ldots, \beta_k\}$. In our case we need the encryption's of only 2^{20} chosen plaintexts, this pool of encryption's contains 21 2^{20} chosen plaintext pairs suitable for the attack. Table 4 compares the growth of the executing time in our attack with Biryukov result [1].

Table 4. Growth in execution time.

r	Our Attack	Biryukov
8	2^{18}	2^{28}
10	2^{126}	2^{28}
12	2^{26}	2^{44}

5. Conclusions and Future Work

In this work, we present a modification of the RC5 encryption algorithm. We have demonstrated the potential effectiveness of timing attack on this modification where the time to encrypt is proportional to the data dependent rotation. This means that the data complexity of cryptanalysis of the modification algorithm is bounded by the probability of a good pair and the capability to detect good pair. However, the analysis clearly demonstrates the susceptibility of some implementations to cryptanalysis if such analysis could be made and suggests that it is vital for designers to aware of cryptographic issues when implementing the modified RC5 algorithm.

References

[1] Biryukov A. and Kushilevitz E., "Improved Cryptanalysis of RC5," *in Proceedings of Advances in Cryptology (Eurocrypt'1998)*, Springer-Verlag,pp. 85-99, 1998.

[2] Kaliski B. S. and Yin Y. L., "On Differential and Linear Cryptanalysis of the RC5 Encryption Algorithm," *in Proceedings of Advances in Cryptology (CRYPTO'95)*, pp. 171-184, 1995.

[3] Knudsen L. R. and Meier W., "Improved Differential Attacks on RC5," *in Proceedings of Advances in Cryptology (CRYPTO'1996)*, Springer Verlag, pp. 216-228, 1996.

[4] Knudsen L. R., Meier W., Preneel B., Rijmen V., and Verdoolaege S., "Analysis Method of Alledged RC4," *in Proceedings of ASIACRYPT'1998*, pp. 327-341, 1998.

[5] Kocher P. C., "Timing Attacks on Implementations of Diffe-Hellman, RSA, DSS, and other Systems," *in Proceedings of Advances in Cryptology (CRYPTO'1996)*, Springer-Verlag, pp. 104-113, 1996.

[6] Rivest R. L., "The RC5 Encryption Algorithm," *Dr. Dobb's Journal*, no. 226, pp. 146-148, January 1995.

[7] Rivest R. L., "The RC5 Encryption Algorithm," *in Proceedings of the Second International Workshop on Fast Software Encryption*, Springer Verlag, New-York, pp. 86-96, December 1994.

[8] Stallings W., *Cryptography and Network Security*, Prentice-Hall, New Jersey, 2003.

Mowafaq Hasan obtained his BSc and Msc from Baghdad University, Iraq in 1980, and his PhD in computer science from University College of Wales, UK in 1989. Currently, he is the head of Information Technology Department, University of Al-Hussein Bin Talal, Jordan. His research interests include object oriented design, data security, and computer and digital image processing.

Hasan Al-Shalabi obtained his BSc and Msc in Computer Engineering from Donetsk Polytechnic Intuitions, Ukraine in 1991, and his PhD in Computer Engineering from Kyiv Polytechnic Intuitions, Ukraine in 1994. Currently, he is the head of Computer Engineering Department, University of Al-Hussein Bin Talal, Jordan. His research interests include e-learning, object oriented programming, data security, computer networks, and computer hardware.